Data-driven Proficiency Profiling

Behrooz Mostafavi
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
bzmostaf@ncsu.edu

Zhongxiu Liu
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
zliu24@ncsu.edu

Tiffany Barnes
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
tmbarnes@ncsu.edu

ABSTRACT

Deep Thought is a logic tutor where students practice constructing deductive logic proofs. Within Deep Thought is a data-driven mastery learning system (DDML), which calculates student proficiency based on rule scores weighted by expert-decided weights in order to assign problem sets of appropriate difficulty. In this study, we designed and tested a data-driven proficiency profiler (DDPP) method in order to calculate student proficiency without expert involvement. The DDPP determines student proficiency by comparing relevant student rule scores to previous students who behaved similarly in the tutor and successfully completed it. This method was compared to the original DDML method, proficiency based on average rule scores, and proficiency based on minimum rule scores. Our testing has shown that while the DDPP has the potential to accurately calculate student proficiency, more data is required to improve it.

Keywords

Data-driven, Tutoring system, Student classification

1. INTRODUCTION

Data-driven methods, methods where each step and calculation is based on analyzing a set of historical data, have been used to great effect to improve individualized computer instruction. They have been used in intelligent tutoring systems to accurately predict student behavior and improve learning outcomes. In contrast to individualized tutoring systems based on developing complex and context specific models of behavior, data-driven systems reduce the need for expert involvement to design the system, and can potentially adapt to new users without refinement of a behavioral model. This is because data-driven systems analyze previous student data in order to model student behavior and determine the best course of outcome in the tutor. Therefore, developing a data-driven intelligent tutoring system is based on gathering data, and developing the methods the system uses to analyze and react to student behavior.

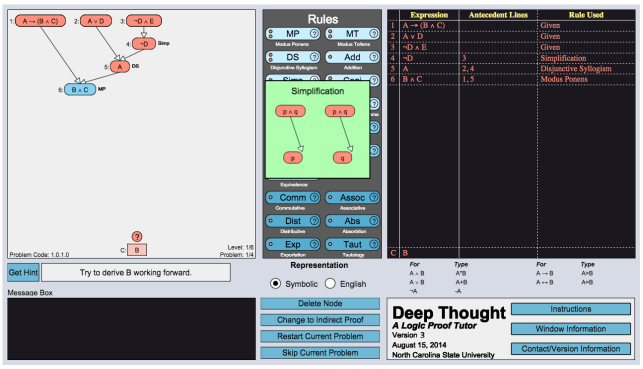

Figure 1: The Deep Thought DT3 logic tutor. Students apply logic rules (axioms) to premises to derive new statements until the conclusion (at the bottom) is justified. The right window displays the proof in standard list format.

We have been incrementally augmenting the Deep Thought logic tutor (Fig. 1) with data-driven methods for formative feedback and problem selection to improve student learning and reduce tutor dropout. Our long term goal is to create an intelligent tutor for logic proof construction that is fully data-driven and can adapt to students learning logic with varying curricular requirements without the need for further expert input. To this end, the next step in our work is to replace the expert-authored assessment parameters built into our problem selection system with a data-driven proficiency calculation that approximates the original system's performance.

Deep Thought utilizes a data-driven mastery learning system (DDML) consisting of 6 strictly ordered levels of proof problems. Each level is split into a higher proficiency track with a lower number of complex problems, and a lower proficiency track with a greater number of simpler problems. The first level of problems are the same for all students, and are used to estimate their initial proficiency. Proficiency is calculated using the knowledge tracing of all rule-application actions taken in the tutor. These action scores are compared to the average score thresholds of corresponding problems solved by past *exemplars* – students who have successfully completed the entire tutor, and have therefore demonstrated sufficient proficiency in the subject matter (Fig. 2).

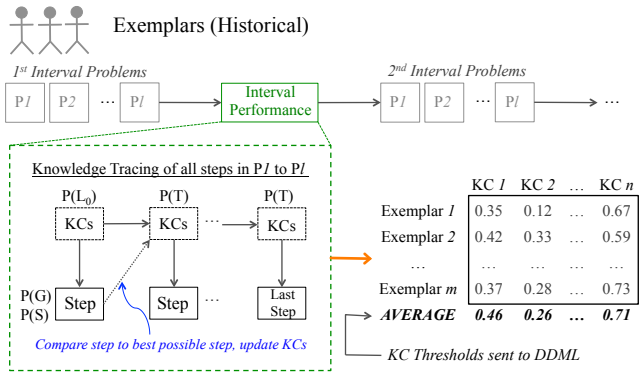

Figure 2: The DDML's threshold builder. Knowledge components (KCs) for each exemplar are updated using action steps from an interval set of tutor problems. The KC score averages at each interval are used as thresholds in the DDML system.

The difference between each action score minus its threshold is weighted by the expert-decided *priorities* of those actions within the level (Eq. 1). The sign of the resulting score determines placement in either the higher (+) or lower (−) proficiency track. On each subsequent level the system will first estimate a student's proficiency and then assign them to the higher or lower proficiency track based upon their prior performance. This system was shown to increase student completion and reduce tutor dropout over unordered and hint-based versions of Deep Thought [10].

Level l End Proficiency =

$$\textbf{sign}\left[\sum_{i=rule_0}^{rule_n}(scoreSign_{l,i} \times rulePriority_{l,i})\right] \quad (1)$$

Since the current DDML system uses expert-decided priorities for each of the rule application actions when calculating a student's proficiency, any new problems or levels added to the system will require expert involvement to determine which rules were prioritized in each new or altered level. This paper describes a study to develop a data-driven method of determining student proficiency that can replace the current expert-decided rule priorities in Deep Thought. This Data-driven Proficiency Profiler (DDPP) uses the clustering of exemplar scores at each level interval for each rule, weighted by primary component importance, to classify exemplars into *types* of student progress through the tutor. New students using the tutor will be assigned to a proficiency track based on comparison to existing types.

The DDPP method is compared alongside proficiency calculations using the minimum rule scores and average rule scores of exemplars, also weighted by primary component importance, to see how these methods compare to each other and to the expert authored system. We hypothesize that the DDPP will perform more accurately than the minimum or average methods of student proficiency classification. This would allow Deep Thought to be used in other classrooms where the pedagogical method and problem-solving ability of the class may be disparate from the current exemplar data from Deep Thought.

Our results show that proficiency calculation using average rule scores performs more accurately than proficiency calculated based on minimum rule scores. In addition, the DDPP method performs more accurately than the average method in some parts of the tutor, while it is less accurate in other parts. Unfortunately, the DDPP system does not yet reach the accuracy of the original system overall in calculating student proficiency. We conclude that more data is required in order for the DDPP to properly approximate the accuracy of the original system's proficiency calculation.

2. RELATED WORK

2.1 Data-driven Tutoring

An early example of a data-driven intelligent tutor is the Cognitive Algebra Tutor[12]. Here the authors introduce an algebra tutor which models student behavior based on the cognitive theory ACT-R and student data gathered from several previous studies. The Cognitive Algebra Tutor was several years and studies into development at this time, and the result is an example of a mostly-realized data-driven tutor. The tutor as it stood improved student performance, and the authors noted that although it over-predicted student performance, it would be improved the more data was collected. However, this system still took a long time and a great deal of expert involvement to design and improve. Conversely, developing a data-driven method of student assessment would reduce this time and effort, since it would be based on analyzing previous data rather than developing and improving on a cognitive model.

Later analyses on the potential benefits, and recommendations, for using data-driven methods to develop intelligent tutoring systems have focused on improving the modelling of student behavior rather than using data to improve on student assessment. Koedinger et al[7] give a very detailed overview on developing data-driven intelligent tutoring systems, and techniques for incorporating data in a useful way. They discuss optimizing the cognitive model using learning factors analysis; fitting statistical models to individual students; modeling student mood and engagement by modeling off-task behaviors, careless errors, and mood; and improving how the tutor selects actions for the student via MDP or POMDP. In a later work[8] the authors compare and contrast current data-driven methods for intelligent tutoring and discuss the potential for these methods to improve MOOCs. They go over the success cases for using data to improve tutors and coursework, in particular cognitive task analysis.

There have been several recent studies that demonstrate the potential for data driven methods to result in tutors that more accurately assess student performance and react to student behavior. Lee and Brunskill[9] examined the benefits and drawbacks to basing model parameters on existing data from individual students in comparison to data from an entire population, specifically as it pertained to the number of practice opportunities a student would require (estimated) to master a skill. The authors estimated that using individualized parameters would reduce the number of practice opportunities a student would need to master a skill. Gonzalez et al.[4] demonstrated a data-driven model which au-

tomatically generated a cognitive and learning model based on previous student data in order to discover what skills students learn at any given time, and when they use skills they have learned. The resulting model predicted student behavior without the aid of previous domain knowledge and performed comparably to a published model.

Data-driven intelligent tutors not only have the potential to more accurately predict student behavior, but interpret why it occurs. For instance, Elmadani et al. [2] proposed using data-driven techniques to detect student errors that occur due to genuine misunderstanding of the concepts (misconception detection). They processed their data using FP-Growth in order to build a set of frequent itemsets which represented the possible misconceptions students could make. The authors were able to detect several misconceptions based on the resulting itemsets of student actions. Fancsali[3] used data-driven methods to detect behaviors that usually detract from a student's experience with an ITS (off-task behavior, gaming the system, etc).

2.2 Cluster-based Classification

Cluster-based classification has several advantages when applied to data-driven tutoring. New educational technologies may reveal unexpected learning behaviors, which may not yet be incorporated in expert-decided classification processes. For example, Kizilec et al. [5] clustered MOOC learners into different engagement trajectories, and revealed several trajectories that are not acknowledged by MOOC designers. In addition, experts classify using their perception of the average students' performance[11] [13]. This perception may be different from the actual participant group. Cluster-based classification methods, however, are able to classify and update classifications based on actual student behaviors.

Moreover, previous studies have shown that personalized tutoring based on cluster-based classification not only helps learning, but improves users' experience. Klasnja-Milicevic et al. [6] gave students different recommendations on learning content based on their classified learning styles. As a result students who used hybrid recommendation features completed more learning sessions successfully, and perceived the tutor as more convenient. Despotovic-Zrakic et al. [1] adapted different course-levels, learning materials, and content in Moodle, an e-learning platform, for students in different clusters. Results showed that students with adapted course design had better learning gain, and a more positive attitude towards the course.

However, the majority of previous work clustered students solely on their overall performance statistics. In contrast, our method clusters students based on their application of specific knowledge components throughout the tutor.

3. METHODS

The Data-driven Proficiency Profiler (DDPP) is a system which calculates student proficiency at the end of each level in Deep Thought based on how a given student performs in comparison to exemplars who employed similar problem solving strategies (see Fig. 3), with rule scores weighted as determined through principal component analysis. Based on how similar exemplars were assigned in subsequent levels, the DDPP can determine the best proficiency level for a new student. In contrast to the DDML system previously employed, this proficiency calculation and rule weighting is entirely data-driven, with no expert involvement. We hypothesize that the DDPP based calculation will perform more accurately when compared to average and minimum methods.

3.1 Data-driven Problem Profiler

We first determined similar problem solving strategies among the exemplars by clustering the exemplars' rule scores (KCs) based on hierarchical clustering. For the initial single-point distance measure we used Euclidean squared distance, while for the hierarchical clustering algorithm we used cluster centroids to determine the distance between individual clusters. As a result each exemplar is assigned to a set of n clusters (where n is equal to the number of KCs), as shown in the table in Fig. 3.

Expert weighting was replaced by principal component analysis (PCA) of the frequency of the rules used for each exemplar for each level, accounting for 95% variance of the results. PCA is typically used to reduce the dimensionality of a data set by determining the most influential factors in the data set. The influence of a given factor is based on how much that factor contributes to the variability in the data. We use PCA analysis on the Deep Thought data set to determine which rules were most important to success in the tutor at each level. Rules which account for 25% of importance and higher are considered most important for completing a level. This percentage was determined through testing, and is the percentage that maximized accuracy. For each rule, its PCA importance value is the new weight for that rule score. Unlike expert authored weights, these rule score weights are based on each rule's importance as determined by the data.

When a new student uses the tutor, the student's rule scores are calculated throughout the level. At the end of each level, the DDPP looks at each student's individual rule score and assigns it to a cluster for that rule. The DDPP then finds which clusters the scores for the most important rules fall into for that level (based on the same PCA based weighting), and then classifies that student into a type based on the set of clusters the student matches (see Fig. 3, right). Finally the system assigns the student to a proficiency track based on data from the matching type of exemplars, and how those exemplars were placed in the next level. The more exemplars we have of a given type, the stronger the prediction we can make for a new student. In the event that a new student doesn't match an existing type in the exemplar data, their proficiency is calculated using the average scores. Average scores are used as a default because, as shown in the results, for most levels it is a better prediction approximation than using the minimum scores.

3.2 DDPP Advantages

In the original system, the student proficiency was determined based on one set of rule thresholds and a set of expert authored weights. However as a result, the system didn't take varying student problem-solving strategies into account. The data is based on students who completed the tutor, who have therefore shown the level of mastery required to successfully complete Deep Thought. However the

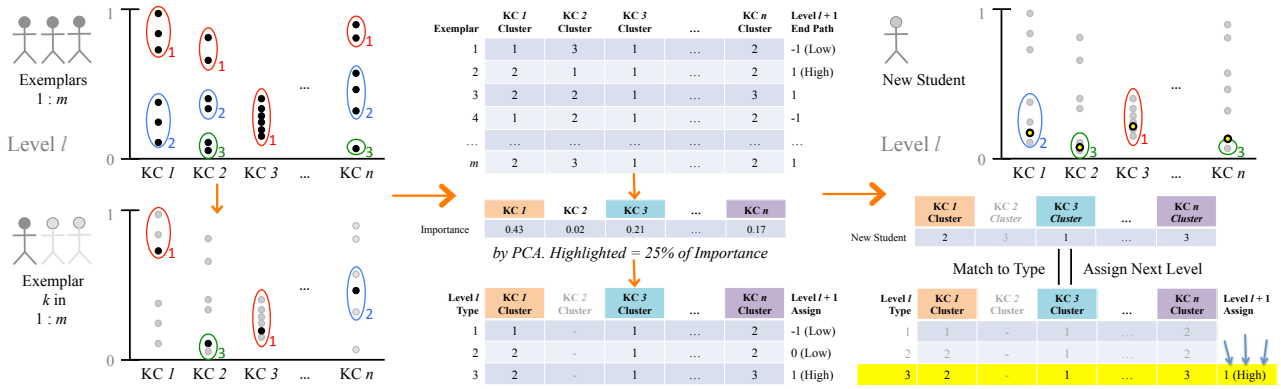

Figure 3: The Data-Driven Proficiency Profiler. (Left) At each level interval, exemplar KC scores are clustered, and exemplars are assigned a cluster for each KC Score. (Center) KCs that make up 25% of importance in the current level are used to assign exemplars to types. (Right) New student scores are assigned to clusters, and compared to existing types to determine next level path.

scores are averaged over all the students at the end of each level. By taking the average of these student scores at this point, we're still assuming only one successful problem solving strategy for completing each level in the tutor. However while most strategies might be the same for earlier levels, there may be a variety of strategies in later levels that can still result in successful completion.

The DDPP method accounts for that possible variety in problem solving methods. In using an unsupervised clustering method, we're able to account for different clusters while not knowing how many clusters there are for each rule. By clustering the scores, we're essentially looking for different strategies that utilize particular rules and determining these strategies based on the student data. Once we determine which strategy a new student is utilizing, we can look to the data again to see how exemplars who employed a similar strategy were placed in the tutor and how they performed, thus determining the best way for the tutor to react to that particular student. Using PCA based weights allows us to weight rule scores based on rule importance as determined by previous students who completed the tutor, rather than expert determination.

3.3 Evaluation

Testing was performed on data collected from two courses using Deep Thought with the DDML system. The first was a Philosophy deductive logic course ($n = 47$) using Deep Thought as a regular assignment over the course of a 15-week semester. The second was a Computer Science discrete mathematics course ($n = 84$), using Deep Thought as a two week assignment during the course's 4-week logic curriculum. From the students in these data, 26 of the Philosophy students (55%) and 50 of the Computer Science students (60%) completed the tutor, and were used as exemplars for the compared methods. By completing all levels in Deep Thought, these students have demonstrated sufficient mastery of the skills needed for introductory proof problem-solving.

By using data from both Computer Science and Philosophy based teaching methods for propositional logic, we expand the range of problem solving strategies analyzed and exemplar types determined. This allows us to test the tutor's performance across different classroom conditions, and determine whether the methods for proficiency path placement are effective for students in different disciplines that use different teaching methods.

The DDML system used the average of exemplar rule scores, weighted by expert-authored end of level rule priorities, to calculate student proficiency. In total there were 19 individual rule actions in Deep Thought on which students were evaluated. Based on the results of this calculation, the DDML system determines whether to send a student on the higher or lower proficiency path in the next level. The system also allowed for the possibility of students switching proficiency paths in situations where the student cannot complete the level on the path they were originally assigned. Because students can switch paths in the middle of a level, we can determine if they finish the current level on the same path they were assigned. If the student did not finish the level on the same proficiency path, it is an indication that the DDML system may have initially assigned the student to the wrong proficiency path. Therefore we can calculate the accuracy of the original system by determining how often students who completed the entire tutor changed proficiency paths throughout. Given $S_{sameTrack}$ as the number of students who finished a level on the same proficiency track, and S_{total} as the total number of students who completed the level, the path prediction accuracy for each level ($LevelAccuracy$) is calculated as follows:

$$LevelAccuracy = \frac{S_{sameTrack}}{S_{total}} \quad (2)$$

The $LevelAccuracy$ for each level is added together to determine the path prediction accuracy. This calculation tells us, for students who completed the entire tutor, how well the original system predicted the paths for them to continue on. This serves as a basis of comparison between the DDPP and the original DDML system.

3.3.1 Minimum & Average

The average rule scores are the set of average scores for each rule in each level. Minimum scores are the smallest scores in the exemplar data set for each rule in each level. This calculation is based on the assumption that if a student scores at least at this minimum for a given rule in that level, the student should be able to perform as well as an exemplar throughout the tutor. The difference between the current DDML system and average score or minimum score based proficiency calculation is that the DDML weighted scores with expert-decided rule priorities, while average or minimum weighted average or minimum scores with PCA-determined weights. Calculating proficiency based on average and minimum scores offers insight into how introducing PCA to students' performance baseline changes the prediction accuracy.

4. RESULTS

The prediction accuracy of the minimum, average, and DDPP methods were calculated for the 76 exemplars from the Philosophy and Computer Science data sets. Ten-fold cross validation was used to train and test the methods across the combined data. We focus on the results of the path prediction accuracy described in section 3.3 as a basis of comparison between the original system, the DDPP, proficiency based on average scores, and proficiency based on base minimum scores. These results are in tables 1, 2, and 3.

4.1 Path Prediction Accuracy

Table 1 shows the path prediction accuracy of the DDML system, the DDPP system, average score assessment, and minimum score assessment across all the students in the Philosophy and CS courses. The original system accuracy was very high, ranging from 75% at the end of level 3 to 88.2% at the end of level 1. The DDPP was somewhat accurate, ranging from 61.8% path prediction accuracy at the end of level 4 to 67.1% path prediction accuracy at the end of level 2. While these accuracies are not nearly as high as in the original system, they are very good considering that, unlike the original system, path prediction in the DDPP is entirely data-driven. It should also be noted that the DDPP was more consistent in its accuracy, only varying by at most 5% between levels (in comparison to the original DDML system, which ranged in accuracy by 9.3%).

Table 1: Path prediction accuracy of the original DDML system, the DDPP system, average score assessment, and minimum score assessment, for both Philosophy and CS students at the end of each level

	Original	DDPP	Average	Minimum
Lvl 1	88.2%	65.8%	65.8%	35.5%
Lvl 2	85.5%	67.1%	73.7%	18.4%
Lvl 3	75.0%	63.2%	60.5%	69.7%
Lvl 4	78.9%	61.8%	64.5%	40.8%
Lvl 5	78.9%	64.5%	59.2%	59.2%

Overall the original system predicted paths more accurately than the DDPP, average, or minimum methods across all levels. The minimum method was least accurate across all levels. In comparison to the average method, the DDPP was more accurate than the average method at the end of levels 3 and 5. The DDPP was equally as accurate as the average method at the end of level 1, and less accurate at the end of levels 2 and 4. However, some of the lower accuracy was likely due to the distribution of exemplars across the two courses. Recall that the CS students made up a higher proportion of the analyzed exemplars than the Philosophy students. Analyzing the path prediction accuracy by the individual course reveals more detail on the path prediction accuracy.

4.2 Philosophy & CS Accuracy

In the case of the Philosophy students, where proportionally fewer of the students were selected as exemplars, the DDPP system was more accurate than the original system on every set of levels except for the end of level 5 (see Table 2). In comparison to the average calculation method, the DDPP was only more accurate at the end of level 3. At the end of levels 1 and 5, the DDPP was as accurate as the average method, and at the end of levels 2 and 4 the DDPP was less accurate.

Table 2: Path prediction accuracy of the original DDML system, the DDPP system, average score assessment, and minimum score assessment, for Philosophy students

	Original	DDPP	Average	Minimum
Lvl 1	76.9%	80.8%	80.8%	23.1%
Lvl 2	65.4%	69.2%	76.9%	19.2%
Lvl 3	50.0%	84.6%	80.8%	38.5%
Lvl 4	65.4%	69.2%	76.9%	30.8%
Lvl 5	53.8%	46.2%	46.2%	26.9%

In the CS course, where proportionally more of the students were selected as exemplars, not only was the original system far more accurate than it was for the entire set of students overall, but the DDPP path accuracy was much worse in some places. However, in comparison to the average method, the DDPP method was only less accurate in level 2. In all other levels the DDPP was either more accurate than the average method (levels 3 and 5) or equally as accurate (levels 1 and 4).

Table 3: Path prediction accuracy of the original DDML system, the DDPP system, average score assessment, and minimum score assessment, for Computer Science students

	Original	DDPP	Average	Minimum
Lvl 1	94.0%	58.0%	58.0%	42.0%
Lvl 2	96.0%	66.0%	72.0%	18.0%
Lvl 3	88.0%	52.0%	50.0%	86.0%
Lvl 4	86.0%	58.0%	58.0%	46.0%
Lvl 5	92.0%	74.0%	66.0%	76.0%

4.3 Discussion

In the original DDML method, the weight of each rule was determined by domain experts. Our results show that when replacing the original weights by weights determined through principal component analysis in the average score method, the prediction accuracy increases for all levels in the philosophy class, but decreases for all levels in the computer

science class. This may because the experts were computer science students and teachers, who prioritized rules with the performance of computer science students in mind. When the real participants were philosophy students, Principal Component Analysis outperformed experts because it prioritized rule based on the performance of the real participants. It's possible that expert involvement may be constrained by the expert's background, whereas a data-driven approach is more flexible when adapting to the diversity of participants.

When comparing the path prediction accuracy of the original method to the DDPP, our result shows that the DDPP calculated student proficiency with more accuracy in the case of the Philosophy students, but less accuracy overall or in the case of the Computer Science students. It is likely that these results are a product of the limited, uncontrolled nature of the dataset. Only 76 exemplars were chosen overall, and of those exemplars a disproportionate number of them were selected from the computer science course. We noticed in the data that the students in the Computer Science course had KC weights that were vastly different than the expert weights. This means the students in the Computer Science course were showing some unorthodox problem solving strategies, particularly in the earlier levels. With enough data and more students with varying strategies, the DDPP could more accurately assign other students who employ different proof solving strategies. However for this limited dataset, it is possible that there were not enough students employing the same unorthodox strategies that a type could be determined.

Table 4: The average number of types found per level during training (exemplars), and the number of students typed during testing (new students). There were a total of 76 students in the data set.

Level	**1**	**2**	**3**	**4**	**5**
Avg. Types Found (Train)	14	13	21	17	26
# Types Matched (Test)	0	4	2	2	10

Table 4 shows the average number of types found in the training dataset, and the number of students matched to a type during testing. While there were several types found in the training step, far fewer students could be matched to a type in the testing step. This would explain the lower accuracy in the DDPP system, as well as why it performed similarly to the average method; it is likely that many of the students in the test set could not be classified into a type, which would result in the DDPP using the calculation based on average scores to determine student proficiency.

That said, the DDPP is still very accurate considering that, in all aspects of proficiency calculation, it is completely data-driven. Its accuracy when applied to the students in the Philosophy class in particular shows the potential for this system to be useful in different classroom conditions. The clustering step at each level produced between 14 and 26 possible types of exemplars to compare students to, compared to what would have been 76 individual students in the original system. This results in a system of proficiency calculation that, given more data, has the potential to calculate student proficiency just as accurately and more efficiently as the original.

5. CONCLUSIONS & FUTURE WORK

We have presented a fully data-driven student proficiency calculator, the Data-driven Proficiency Profiler (DDPP). The DDPP clusters exemplar student data into types, attempts to classify new students into one of the exemplar types, and calculate proficiency based on exemplars who employed similar problem strategies. We hypothesized that the DDPP would be more accurate than proficiency calculated using average scores or minimum scores. Instead, our results showed that the DDPP performed about as well as the average method overall, and did not approximate the accuracy of the original system. However our data set was very limited, and the high accuracy the DDPP achieved for the Philosophy students shows this system has potential once more data can be acquired.

In the future, we would like to be able to test this system with more data. The more students use the system, the greater the data set we will be able to use and the more conclusions we will be able to draw on the qualities of the DDPP system. In particular we will analyze in greater detail the types found on each level and the differences between each type in terms of problem solving strategy. We can also determine the importance, in depth, of certain rules to each level and the problems within it based on student problem solving strategies. Our final step is to implement the DDPP into Deep Thought and use it to direct students through the levels. Implementing the DDPP into Deep Thought will allow us to test whether, ultimately, the DDPP is an accurate, data-driven proficiency calculation.

6. ACKNOWLEDGEMENTS

This material is based on work supported by the National Science Foundation under Grants 1432156 and 0845997.

7. REFERENCES

[1] M. Despotovic-Zrakic, A. Markovic, Z. Bogdanovic, D. Barac, and S. Krco. Providing adaptivity in moodle lms courses. *Educational Technology Society*, 15(1):326–338, 2012.

[2] M. Elmadani, M. Mathews, and A. Mitrovic. Data-driven misconception discovery in constraint-based intelligent tutoring systems. In *Proceedings of the 20th International Conference on Computers in Education (ICCE)*, pages 26–20, 2012.

[3] S. E. Fancsali. Causal Discovery with Models: Behavior, Affect, and Learning in Cognitive Tutor Algebra. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*, pages 28–35, 2014.

[4] J. P. Gonzalez-Brenes and J. Mostow. What and when do students learn? Fully data-driven joint estimation of cognitive and student models. In *Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013)*, pages 236–239, 2013.

[5] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In

Proceedings of the 3rd international conference on learning analytics and knowledge, pages 170–179, 2013.

[6] A. Klasnja-Milicevic, B. Vesin, M. Ivanovic, and Z. Budimac. E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers Education*, 56(3):885–899, 2011.

[7] K. R. Koedinger, E. Brunskill, R. S. Baker, E. A. McLaughlin, and J. Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.

[8] K. R. Koedinger, E. A. McLaughlin, and J. C. Stamper. Data-driven Learner Modeling to Understand and Improve Online Learning: MOOCs and technology to advance learning and learning research (Ubiquity symposium). In *Ubiquity 2014*. 2014.

[9] J. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Proceedings of the 5th International Conference on Educational Data Mining (EDM 2012)*, pages 118–125, 2012.

[10] B. Mostafavi, M. Eagle, and T. Barnes. Towards Data-driven Mastery Learning. In *To appear in Proc. Learning, Analytics, and Knowledge (LAK 2015)*.

[11] E. V. Perez, L. M. R. Santos, M. J. V. Perez, J. P. de Castro Fernandez, and R. G. Martin. Automatic classification of question difficulty level: Teachers' estimation vs. students' perception. In *Proceedings of the IEEE Frontiers in Education Conference*, pages 1–5, 2012.

[12] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic BulletinReview*, 14(2):249–255, 2007.

[13] G. van de Watering and J. van der Rijt. Teachers and students perceptions of assessments: A review and a study into the ability and accuracy of estimating the difficulty levels of assessment items. *Educational Research Review*, 1(2):133–147, 2006.